

Claim Amendment Summary**Claims pending**

- At time of the Action: Claims 1-33.
- After this Response: Claims 1-2 and 10-60.

Cancelled claims: 3-9.**Amended claims:** 1 and 14.**New claims:** 34-60.

1. **(CURRENTLY AMENDED)** A method of parsing an Extensible
9 Markup Language (XML) data stream comprising:

10 defining a plurality of states, individual states being associated with
11 individual elements of an XML data stream;

12 associating one or more rules with each state;

13 receiving an XML data stream;

14 evaluating the XML data stream against one or more of the rules for
15 individual elements contained in the XML data stream; and

16 processing disregarding only those associated portions of the XML data
17 stream that do not violate if any of the rules that are associated with those portions
18 are violated.

21 2. **(ORIGINAL)** The method of claim 1, wherein the one or more rules
22 relate to a schema of the XML data stream.

24 3-9. **(CANCELLED)**

1 10. **(ORIGINAL)** The method of claim 1 further comprising defining
2 one or more rules that relate to an element's contents.

3
4 11. **(ORIGINAL)** The method of claim 10, wherein said one or more
5 rules that relate to an element's contents define which elements can be contained
6 within other elements.

7
8 12. **(ORIGINAL)** The method of claim 11, wherein if a rule that defines
9 which elements can be contained within other elements is violated, disregarding
10 associated portions of the XML data stream until a close tag is received for an
11 element that violates the rule.

12
13 13. **(ORIGINAL)** A computer-readable medium having a program
14 thereon which, when executed by a computer, performs the steps of claim 1.

15
16 14. **(CURRENTLY AMENDED)** A method of parsing an Extensible
17 Markup Language (XML) data stream comprising:

18 defining a schema module that is associated with an HTTP request type that
19 is received from a client, the schema module having a function that determines
20 whether an XML data stream conforms to a given schema that is associated with
21 the HTTP request type;

22 evaluating an XML data stream with the schema module; and

23 processing disregarding a only those portions of the XML data stream that
24 if it does not conform to the given schema.

1 15. **(ORIGINAL)** The method of claim 14, wherein said defining of the
2 schema module comprises defining a plurality of schema modules, individual
3 schema modules being associated with different HTTP request types.

4

5 16. **(ORIGINAL)** The method of claim 14, wherein said function
6 determines whether there are any unauthorized elements that appear in a client's
7 request.

8

9 17 **(ORIGINAL)** The method of claim 14, wherein said function
10 determines whether there are any unauthorized elements that appear in a client's
11 request; said disregarding comprising disregarding said XML data stream portion
12 until a close tag is received for an unauthorized element.

13

14 18. **(ORIGINAL)** The method of claim 14, wherein said HTTP request
15 type comprises a WebDAV request type.

16

17 19. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
18 request type comprises a PROPFIND request.

19

20 20. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
21 request type comprises a PROPPATCH request.

22

23 21. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
24 request type comprises a SEARCH request.

1 22. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
2 request type comprises one of a LOCK and UNLOCK request.

3
4 23. **(ORIGINAL)** A computer-readable medium having a program
5 thereon which, when executed by a computer, performs the steps of claim 14.

6
7 24. **(ORIGINAL)** An Extensible Markup Language (XML) parsing
8 system comprising:

9 a parser configured to receive an XML data stream and generate a series of
10 calls as it parses the XML data stream;

11 a node factory communicatively associated with the parser and configured
12 to receive the parser's calls and responsive thereto construct a representation of the
13 XML data stream that the parser is parsing; and

14 a schema module communicatively associated with the node factory and
15 configured to evaluate the node factory's representation of the XML data stream
16 and determine whether it conforms to a known schema.

17
18 25. **(ORIGINAL)** The parsing system of claim 24, wherein said parsing
19 system comprises a plurality of schema modules, each schema module being
20 associated with a different known schema.

21
22 26. **(ORIGINAL)** The parsing system of claim 24, wherein the schema
23 module corresponds to an HTTP request type.

24
25 27. **(ORIGINAL)** The parsing system of claim 24, wherein said parsing
system comprises a plurality of schema modules, each schema module being

1 associated with a different known schema and corresponding to a different HTTP
2 request type.

3
4 28. (ORIGINAL) The parsing system of claim 27, wherein at least one
5 of the different HTTP request types is a WebDAV request.

6
7 29. (ORIGINAL) The parsing system of claim 24, wherein the schema
8 module is configured to ignore an XML data stream portion that does not conform
9 to the known schema.

10
11 30. (ORIGINAL) An Extensible Markup Language (XML) parsing
12 system comprising:

13 a collection of schema modules, each of which being configured to evaluate
14 a different schema that is associated with an XML data stream; and

15 a plurality of states associated with each schema module, individual states
16 of a schema module defining a schema requirement relating to a particular element
17 that is evaluated by that schema module.

18
19 31. (ORIGINAL) The parsing system of claim 30, wherein each
20 schema module is associated with a different HTTP request and is configured to
21 evaluate a schema that is associated with the HTTP request with which it
22 is associated.

23
24 32. (ORIGINAL) The parsing system of claim 31, wherein at least one
25 of the HTTP requests is a WebDAV request.

1 33. **(ORIGINAL)** The parsing system of claim 31, wherein each of the
2 HTTP requests is a WebDAV request.

3 34. **(NEW)** A method of parsing an Extensible Markup Language
4 (XML) data stream comprising:

5 defining a plurality of states, individual states being associated with
6 individual elements of an XML data stream;

7 associating one or more rules with each state;

8 receiving an XML data stream;

9 evaluating the XML data stream against one or more of the rules for
10 individual elements contained in the XML data stream; and

11 disregarding associated portions of the XML data stream if any of the rules
12 that are associated with those portions are violated, the disregarded portions of the
13 XML data stream representing at least one error in the XML data stream.

14
15 35. **(NEW)** The method of claim 34, wherein the request type is a
16 WebDAV request type.

17
18 36. **(NEW)** The method of claim 35, wherein the WebDAV request type
19 is a PROPFIND request.

20
21 37. **(NEW)** The method of claim 35, wherein the WebDAV request type
22 is a PROPPATCH request.

23
24 38. **(NEW)** The method of claim 35, wherein the WebDAV request type
25 is a SEARCH request.

1
2 **39. (NEW) The method of claim 35, wherein the WebDAV request type**
3 **is one of a LOCK and UNLOCK request.**

4
5 **40. (NEW) A method of parsing an Extensible Markup Language**
6 **(XML) data stream comprising:**

7 **defining a plurality of states, individual states being associated with**
8 **individual elements of an XML data stream, wherein the defining of the plurality**
9 **of states comprises defining one or more schema modules that are configured to**
10 **track one or more states of the XML data stream;**

11 **associating one or more rules with each state;**

12 **receiving the XML data stream;**

13 **evaluating the XML data stream against one or more of the rules for**
14 **individual elements contained in the XML data stream, wherein the evaluating**
15 **comprises using the one or more schema modules to evaluate the XML data**
16 **stream against one or more schema-based rules; and**

17 **disregarding associated portions of the XML data stream if any of the rules**
18 **that are associated with those portions are violated.**

19
20 **41. (NEW) The method of claim 40, wherein each schema module is**
21 **associated with at least one request type that defines the XML data stream.**

22
23 **42. (NEW) The method of claim 41, wherein the request type is a**
24 **WebDAV request type.**

1 43. (NEW) The method of claim 42, wherein the WebDAV request type
2 is a PROPFIND request.

3
4 44. (NEW) The method of claim 42, wherein the WebDAV request type
5 is a PROPPATCH request.

6
7 45. (NEW) The method of claim 42, wherein the WebDAV request type
8 is a SEARCH request.

9
10 46. (NEW) The method of claim 42, wherein the WebDAV request type
11 is one of a LOCK and UNLOCK request.

12
13 47. (NEW) The method of claim 40 further comprising defining one or
14 more rules that relate to an element's contents.

15
16 48. (NEW) The method of claim 47, wherein said one or more rules that
17 relate to an element's contents define which elements can be contained within
18 other elements.

19
20 49. (NEW) The method of claim 48, wherein if a rule that defines which
21 elements can be contained within other elements is violated, disregarding
22 associated portions of the XML data stream until a close tag is received for an
23 element that violates the rule.

24
25 50. (NEW) A computer-readable medium having a program thereon
 which, when executed by a computer, performs the steps of claim 40.

1
2 **51. (NEW) A method of parsing an Extensible Markup Language**
3 **(XML) data stream comprising:**

4 **defining a plurality of states, individual states being associated with**
5 **individual elements of an XML data stream, wherein the defining of the plurality**
6 **of states comprises defining one or more schema modules that are configured to**
7 **track one or more states of the XML data stream each schema module being**
8 **associated with at least one request type that defines the XML data stream;**

9 **associating one or more rules with each state;**

10 **receiving the XML data stream;**

11 **evaluating the XML data stream against one or more of the rules for**
12 **individual elements contained in the XML data stream; and**

13 **disregarding associated portions of the XML data stream if any of the rules**
14 **that are associated with those portions are violated.**

15
16 **52. (NEW) The method of claim 51, wherein the request type is a**
17 **WebDAV request type.**

18
19 **53. (NEW) The method of claim 52, wherein the WebDAV request type**
20 **is a PROPFIND request.**

21
22 **54. (NEW) The method of claim 52, wherein the WebDAV request type**
23 **is a PROPPATCH request.**

24
25 **55. (NEW) The method of claim 52, wherein the WebDAV request type**
is a SEARCH request.

1
2 56. (NEW) The method of claim 52, wherein the WebDAV request type
3 is one of a LOCK and UNLOCK request.

4
5 57. (NEW) The method of claim 51 further comprising defining one or
6 more rules that relate to an element's contents.

7
8 58. (NEW) The method of claim 57, wherein said one or more rules that
9 relate to an element's contents define which elements can be contained within
10 other elements.

11
12 59. (NEW) The method of claim 58, wherein if a rule that defines which
13 elements can be contained within other elements is violated, disregarding
14 associated portions of the XML data stream until a close tag is received for an
15 element that violates the rule.

16
17 60. (NEW) A computer-readable medium having a program thereon
18 which, when executed by a computer, performs the steps of claim 51.